

Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) EP 1 091 283 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
11.04.2001 Bulletin 2001/15

(51) Int. Cl.⁷: G06F 3/06

(21) Application number: 00305256.0

(22) Date of filing: 21.06.2000

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 30.09.1999 JP 27967899

(71) Applicant: FUJITSU LIMITED
Kawasaki-shi, Kanagawa 211-8588 (JP)

(72) Inventors:
• Kaiya, Norikazu,
c/o Fujitsu Aichi Eng. Limited,
Aichi 461-0004 (JP)

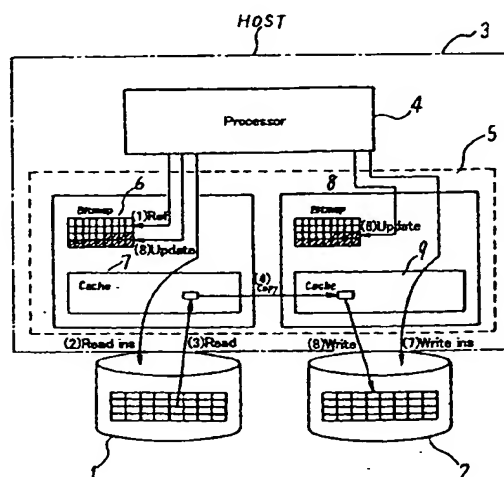
• Onda, Yasuhiro,
c/o Fujitsu Aichi Eng. Limited,
Aichi 461-0004 (JP)
• Kato, Tadaomi,
c/o Fujitsu Limited,
Kawasaki-shi, Kanagawa, 211-8588 (JP)

(74) Representative:
Stebbing, Timothy Charles et al
Haseltine Lake & Co.,
Imperial House,
15-19 Kingsway
London WC2B 6UD (GB)

(54) Copying method between logical disks, disk-storage system and program for the same

(57) This invention relates to a copying method, disk storage system and program for copying data from one logical disk to another logical disk, and makes immediate access in response to a copy command possible. Copying from logical disk (1) (copy source) to logical disk (2) (copy destination) is performed for each area in the copy range, and when there is access, copying is interrupted. When performing update access of an uncopied area in the copy source (1), the area is copied from the copy source (1) to the copy destination, and then the area is updated (S23). When performing reference access of an uncopied area in the copy destination (2), the corresponding area in the copy source (1) is referenced. When performing update access of an uncopied area in the copy destination (2), that area in the copy destination is updated, and copying is prohibited (S13). In this way, it is possible to access the copy source (1) and copy destination (2) while copying is in progress, and it is possible to immediately regard copying as completed by a copy instruction.

FIG. 1



EP 1 091 283 A2

Description

[0001] This invention relates to a copying method between logical disks, disk storage system and a program for copying between logical disks.

[0002] A logical disk is a disk that appears to be unrelated to physical restrictions imposed by a physical disk. A logical disk can be constructed from a plurality of physical disk devices, or from one physical disk with a plurality of logical disks. The kind of disk used is determined by the parameters of capacity and performance.

[0003] A typical system is called a RAID (for example 0, 1, 0+1, 3, 5) rank or RAID group.

[0004] Copying between logical disks is used as a backup for saving data as well as other uses. For example, executing a test while continuing work. In this case, the logical disk that is used for work is copied and the copy-destination logical disk is used by the test system. By doing this, the work being performed using the copy-source logical disk is not affected even if data are damaged by the test system that uses the copy-destination logical disk.

[0005] Also, a logical disk can be used in online work by copying the logical disk used in online work and using the copy-destination logical disk in batch processing. By doing this, the performance of the online work that uses the copy-source logical disk is not affected by the batch processing.

[0006] Many cases such as these which require copying between logical disks are increasing. However, work must be stopped while copying the logical disk. For example, it can take one hour to copy 9 Gbytes. In 24-hour continuous operating conditions of recent years, it is desirable to make the work stop time as close to zero as possible by making the copy time as close to zero as possible.

[0007] Fig. 13 to Fig. 16 are drawings explaining the prior art. Fig. 13 and Fig. 14 explain the prior separate dual volume system, Fig. 15 explains the prior log structured file system, and Fig. 16 explains the prior concurrent copy system.

[0008] The following methods are known as methods for copying logical-disk data in a prior RAID device.

(1) Separate Dual Volume System

[0009] In a system comprising a host 90 and RAID device 91 as shown in Fig. 13, first a copy instruction is sent from the host 91, as shown in Fig. 14, and the copy-destination volume (logical disk) is prepared (defined). Next, in response to an "Establish Pair" instruction, copying begins from a primary (copy-source logical disk) 93 to a secondary (copy-destination logical disk) 94 (to create a dual state). Establishment of a pair is completed after copying is finished (after the dual state is created). After this dual state is created, an update process request from the host 90 is reflected by the primary (copy-source logical disk) 93 and second-

ary (copy-destination logical disk) 94. By issuing an instruction to separate the dual pairs, copying of the logical volume at this state is complete. Now, as shown in Fig. 13, it is possible for different hosts (applications) 90, 95 to use the primary (copy-source logical disk) 93 and secondary (copy-destination logical disk) 94.

[0010] As shown in Fig. 14, the primary (copy-source logical disk) 93 can be accessed during copying, however, the secondary (copy-destination logical disk) cannot be accessed during copying.

(2) Log Structured File System

[0011] As shown in Fig. 15, logical disks A, B, C have been defined by the host. Each data block is stored in one physical disk 96. Each of the logical disks is defined by a data pointer of the physical disk 96. Copying from a logical disk A to a logical disk B, is performed by setting the pointer of the logical disk B as pointer of the logical disk A.

[0012] Moreover, update data B2 for the logical disk B are stored in the physical disk separately from the original data. Also, by updating the pointer of the logical disk B to the pointer of the update data B2, it is possible to update the logical disk B without changing the original data A1 to A6 of the logical disk A.

[0013] This method is not followed by copying of actual data, and by setting pointers, the copy operation can be completed immediately.

(3) Concurrent Copy System

[0014] As shown in Fig. 16, when copying from the copy-source disk 93 to the copy-destination disk 94, the host 90 reads the data to be copied from the copy-source disk (A), and with a data transfer function, writes the data to the copy-destination disk 94. The bitmap of the copied address is set as copied.

[0015] When there is an update request for an uncopied area of the copy-source disk 93 during copying, that data of the copy-source disk 93 is stored away in a side file 92 as shown by mark (1). That address portion of the bitmap is set as being stored away in a side file as shown by mark (2). The copy-source disk 93 is also updated as shown by mark (3). When copying data that have been stored away in the side file 92, the data to be copied are read from the side file 92 and, with the data transfer function, are written to the copy-destination disk 94 as shown by mark (4). This method makes it possible for the host to reference and update the copy-source disk 93 during copying.

[0016] The prior art, however, had the following problems.

(1) In the separate dual volume system shown in Fig. 13 and Fig. 14, it is not possible to access the copy-source disk during copying. Therefore, to access the copy-source disk it is necessary to wait

until copying is finished. For example, copying 9 Gbytes takes about 1 hour, so it is necessary to wait 1 hour. Therefore operation must be performed on a schedule.

(2) In the log structured system shown in Fig. 15, it is possible to access both the copy-source logical disk and copy-destination logical disk with a copy instruction, however, since actual data are not copied, the advantages of a dual system where data are copied on separate disks cannot be realized. In other words, damage to the copy-source logical disk is also damage to the copy-destination logical disk, so when the copy-destination physical disk is damaged, it is also not possible to use the copy-source logical disk. Moreover, accessing the copy-destination physical disk, also means accessing the copy-source physical disk, so it is not possible to obtain high-speed performance from using separate disks.

(3) In the concurrent copy system shown in Fig. 16, access of the copy-source logical disk is also possible during copying, however, in order to access the copy-destination logical disk, it is necessary to wait until copying is finished.

[0017] It is a consideration of the present invention to provide a logical-disk copying method, disk-storage system and program that allow access immediately in the copy complete state according to a copy instruction even though copying real data.

[0018] Another consideration of this invention is to provide a logical-disk copying method, disk-storage system and program for making it possible to access a copy-destination logical disk even during copying.

[0019] A further consideration of this invention is to provide a logical-disk copying method, disk-storage system and program for determining whether or not the access area has been copied and for making it possible to access the copy-destination logical disk.

[0020] According to a first aspect of the invention there is provided a method for copying data from a copy-source logical disk to a copy-destination logical disk according to a copy instruction. This copying method comprises a step of copying data, in units of each area divided in the copy-source logical disk, from the copy-source logical disk to the copy-destination logical disk; a step of interrupting the copy process in response to an access request for the copy-source logical disk or the copy-destination logical disk during copying; and a step of processing said access request.

[0021] Preferably, the access processing step comprises a step of updating a target access area of the copy-source logical disk after the target access area has been copied to the copy-destination logical disk when said request is a request for updating an uncopied area of the copy-source logical disk; a step of reading the target access area of the copy-source logical disk when said request is a request for referring an uncopied

area of the copy-destination logical disk; and a step of updating the target access area of the copy-destination logical disk and prohibiting copying of the target access area when said request is a request for updating of uncopied area of the copy-destination logical disk.

[0022] In this way, the system returns immediately a copy complete reply according to a copy instruction, and receives an access request and performs reference and update access of the copy-source or copy-destination logical disk while copying actual data. In order to do this, the copy-source logical disk is divided into a plurality of areas and copied to the copy-destination logical disk in units of the area divisions. When there is an access request during copying, it interrupts the copy operation and executes the access request.

[0023] The access process may be changed depending on whether the target access area is a copied area or uncopied area. In other words, when performing reference or update access of a copied area, since copying is finished, access is allowed. On the other hand, when accessing an uncopied area, the data when the copy instruction (is issued) are saved and then access is allowed. In other words, when performing reference access of an uncopied area of the copy-source logical disk, since there is no change in data, access is allowed as is. When performing update access of an uncopied area of the copy-source logical disk, the uncopied data before the update would be lost if update is performed as is, so the target access area of the copy-source logical disk is copied to the copy-destination logical disk, thereby the data before the update are moved to the copy-destination logical disk. Then, the target access area of the copy-source logical disk is updated. In this way, it is possible to copy data before an update when there is a copy instruction.

[0024] Furthermore, when performing reference access of an uncopied area of the copy-destination logical disk, the data of that area of the copy-source logical disk are read since copying is not yet finished. By doing this, it is possible to reference the copy data of that area even when the area is an uncopied area of the copy-destination logical disk, and it becomes possible to reference data when there is a copy instruction even for the copy-destination logical disk.

[0025] When performing update access of an uncopied area of the copy-destination logical disk, the copy-destination logical disk is updated with update data. Also, copying of that area is prohibited. In this way, it is possible to prevent update data of that area of the copy-destination logical disk from being changed by copying.

[0026] The access step may further comprise a step of referencing the target access area of the copy-source logical disk when performing reference access of the copy-source logical disk, and a step of referencing or updating the target access area of the copy-destination logical disk when accessing a copied area of the copy-destination logical disk.

[0027] The step of referencing an uncopied area

may comprise a step of copying the target access area of the referenced copy-source logical disk to the copy-destination logical disk after the target access area of the copy-source logical disk has been referenced. By doing this, it becomes possible to copy together with refer-

[0028] The copying step may comprise a step of referencing control information that indicates the copying conditions of the areas, and setting the next copy area, and a step of updating the control information of the copy area after copying of the copy area has finished. Also, the access step comprises a step of referencing the control information and determining whether the target access area is a copied area or uncopied area.

[0029] Since there is control information, it is possible to prevent an area from being copied again even if the data of the target access area are copied first. Moreover, since the control information is used to determine whether or not the target access area is a copied area or uncopied area, it is possible to accurately determine whether the access area is a copied area or uncopied area.

[0030] The access step may further comprise a step of updating the control information of the target access area after the target access area of the copy-source logical disk has been copied to the copy-destination logical disk.

[0031] In this way, it is possible for the control information to accurately indicate the copying condition through access processing even if copying is performed in advance.

[0032] The copying step may comprise a step of referencing the control information that indicates the copying condition of each of the areas of the copy-source logical disk and setting the next copy area, a step of reading the target area of the copy-source logical disk in memory, a step of updating the control information after the target area has been read, and a step of writing the target area of the memory to the copy-destination logical disk.

[0033] By reading the target area in memory, it becomes possible to access the copy-source logical disk by updating the control information.

[0034] The copying step may additionally comprise a step of referencing a bitmap indicated by a flag that indicates the copying condition of each area and setting the next copy area, and a step of updating the copy area flag of the bitmap.

[0035] Since the control information is expressed as a bitmap, it is possible to reduce the amount of memory occupied by the control information.

[0036] The copying step may further comprise a step of notifying when copying is finished according to the copy instruction. By doing this, it is possible to immediately know when the access source can be accessed.

[0037] According to a second aspect of the present invention there is provided a disk storage system com-

prising: one logical disk that is constructed with one physical disk, another logical disk that is constructed with another physical disk, and a disk control circuit (controller) for accessing the data of a specified logical disk according to an access instruction, and copying the data in units of area divisions of the copy-source logical disk to the copy-destination logical disk according to a copy instruction.

[0038] When performing update access of an uncopied area of the copy-source logical disk, this disk control circuit updates the target access area of the copy-source logical disk after it has been copied to the copy-destination logical disk. When performing reference access of an uncopied area of the copy-destination logical disk, the disk control circuit reads the target access area of the copy-source logical disk. When performing update access of an uncopied area of the copy-destination logical disk, the disk control circuit updates the target access area of the copy-destination logical disk, and prohibits copying of the target access area.

[0039] As mentioned above, by doing this, it becomes possible to access the copy-source and copy-destination disks, and in accordance to a copy instruction, the host is able to immediately access the copy-destination and copy-source logical disks. In addition, it is possible to avoid the copy-wait state.

[0040] It is preferred that the disk-storage system of the invention has a memory for storing the control information that indicates the copying condition of the area of the copy-source logical disk, and the control circuit references the control information stored in this memory and determines whether the target access area is a copied area or an uncopied area.

[0041] According to a third aspect of the present invention there is provided a program comprising: information (program code) for copying data, in units of area divisions of the copy-source logical disk, to the copy-destination logical disk, and information for interrupting the copying process and performing the access process when accessing the copy-source and copy-destination logical disks during the copying process. For performing update access of an uncopied area of the copy-source logical disk, this information for performing the access process comprises information for updating the target access area of the copy-source logical disk after the target access area of the copy-source logical disk has been copied to the copy-destination logical disk. For performing reference access of an uncopied area of the copy-destination logical disk, this information for performing the access process comprises information for referencing the target access area of the copy-source logical disk. When performing update access of an uncopied area of the copy-destination logical disk, this information for performing the access process comprises information for updating the target access area of the copy-destination logical disk and prohibiting copying of the access area.

[0042] The program is preferably stored on a stor-

age medium, which may also contain information to be copied.

[0043] A description of the invention will now be given, by way of example, with reference to the accompanying drawings, in which:

Fig. 1 is a block diagram of a disk control device of one embodiment of this invention.

Fig. 2 is a flowchart of a copying process of one embodiment of this invention.

Fig. 3A to Fig. 3D are drawings explaining a bitmap in Fig. 1.

Fig. 4 is a flowchart of a reference process during copying of one embodiment of this invention.

Fig. 5 is a flowchart of an update process during copying of one embodiment of this invention.

Fig. 6 is a drawing explaining an access pattern during copying.

Fig. 7 is a drawing explaining a reference operation of a copied portion of a copy source of one embodiment of this invention.

Fig. 8 is a drawing explaining an update operation of a copied portion of the copy source of one embodiment of this invention.

Fig. 9 is a drawing explaining a reference operation of an uncopied portion of the copy source of one embodiment of this invention.

Fig. 10 is a drawing explaining an update operation of an uncopied portion of a copy destination of one embodiment of this invention.

Fig. 11 is a drawing explaining an update operation of an uncopied portion of the copy source of one embodiment of this invention.

Fig. 12 is a drawing explaining an update operation of an uncopied portion of the copy destination of one embodiment of this invention.

Fig. 13 is diagram showing a prior separate dual volume system.

Fig. 14 is a drawing explaining the operation of the prior separate dual volume system.

Fig. 15 is a drawing explaining a prior log structured system.

Fig. 16 is a drawing explaining a prior concurrent copy system.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0044] Fig. 1 is a block diagram of a RAID device of one embodiment of this invention, Fig. 2 is a flowchart of a copying process of Fig. 1, and Fig. 3 explains a bitmap memory of Fig. 1.

[0045] As shown in Fig. 1, a logical disk 1 is one or more physical disks. A magnetic disk or optical disk can be used as the physical disk. A logical disk 2 is one or more physical disks. A disk controller 3 accesses the logical disks 1 and 2 according to instructions from a host.

[0046] The disk controller 3 comprises a processor (CPU) 4 and memory (main storage memory) 5. The memory 5 has a cache area (called cache memory) 7 for the logical disk 1, and a cache area (called cache memory) 9 for the logical disk 2. Also, the memory 5 has a bitmap area (called bitmap) 6 for controlling the copy status of the logical disk 1, and a bitmap area (called bitmap) 8 for controlling the copy status of the logical disk 2.

[0047] In other words, as shown in Fig. 3, the bitmaps 6, 8 assign one bit to each data block in the copy range of the logical disks 1, 2, and control the copy status of the data blocks by assigning "1" to uncopied blocks and assigning "0" to copied blocks. A copy program is used as the firmware of the processor 4. In addition, it uses the bitmaps 6, 8 to copy the copy range of the logical disks in units of data blocks.

[0048] The copying process will be explained with reference to the flowchart of the copying process shown in Fig. 2. Here, the logical disk 1 is a copy source, and the logical disk 2 is a copy destination.

[0049] (S1) At the start of copying, the host gives a copy instruction to the disk controller 3. After the processor 4 receives the copy instruction, it notifies the host that copying is complete. This makes it possible for the host to access the logical disks 1, 2.

[0050] (S2) The processor 4 sets the update information of the copy-destination bitmap 8 to not-updated. As shown in Fig. 3B, each bit of the bitmaps is set to "1", indicating it is uncopied.

[0051] (S3) The processor 4 references the copy-source bitmap 6 and acquires the uncopied-block information (1). In addition, the processor 4 instructs that target blocks in the copy-source logical disk 1 be read (2). By doing this, the target data blocks of the copy-source logical disk 1 are stored in the copy-source cache memory 7. This is called "Staging" (3).

[0052] (S4) The processor 4 copies the uncopied data blocks, that were read from the copy-source cache memory 7, to the copy-destination cache memory 9.

[0053] (S5) The processor 4 sets the update block information of the copy-source bitmap 6 and the copy-destination bitmap 8 to "0", indicating that update was completed. Fig. 3C shows this state. In addition, the processor 4 instructs that the uncopied data blocks that were copied into the cache memory 9 be written to the copy-destination logical disk 2 (7). By doing this, the uncopied data blocks that were copied into the cache memory 9 are written to the logical disk 2. This is called "Write Back" (8).

[0054] (S6) The processor 4 references the update block information of the copy-source bitmap 6 and determines whether there are any data blocks remaining that should be copied. When there are data blocks remaining that should be copied, the process returns to step S3. On the other hand, if there are no data blocks remaining that should be copied, then copying ends.

[0055] When there is an I/O request interrupt for the

copy-source or copy-destination logical disks 1, 2 between steps S3 to S7 (area indicated by the arrows in Fig. 2), the copying process is interrupted, and that access process is performed (described later using Fig. 4 to Fig. 12).

[0056] Fig. 4 is a flowchart of a process for referencing the logical disks during copying. Fig. 5 is a flowchart of a process for updating the logical disks during copying. Fig. 6 explains an access pattern for accessing the logical disks during copying, while Fig. 7 to Fig. 12 explain that operation. Fig. 7 is a drawing explaining the reference operation of a copied portion of the copy source. Fig. 8 is a drawing explaining an update operation of a copied portion of the copy source. Fig. 9 is a drawing explaining a reference operation of an uncopied portion of the copy source. Fig. 10 is a drawing explaining an update operation of an uncopied portion of the copy destination. Fig. 11 is a drawing explaining an update operation of an uncopied portion of the copy source. Fig. 12 is a drawing explaining an update operation of an uncopied portion of the copy destination.

[0057] Fig. 4 will be used to explain the reference process during copying.

[0058] (S10) When the processor 4 detects a reference instruction (read), it determines whether actual copying is in progress. When actual copying is not in progress, the process advances to a normal reference process of step S15.

[0059] (S11) When actual copying is in progress, the processor 4 determines whether the reference instruction is a reference request for the copy source. When the reference instruction is a reference request for the copy source (access pattern 1, 3 in Fig. 6), the process advances to the normal reference process of step S15.

[0060] (S12) When the reference instruction is not the reference request for the copy source, the request is a reference request for the copy destination. The processor 4 references the bitmap 8 and determines whether or not the requested reference area is an uncopied portion. When the request is not a reference request for an uncopied portion of the copy destination, or in other words, when the request is a reference request for a copied portion of the copy destination (access pattern 5 in Fig. 6), the process advances to the normal reference process of step S15.

[0061] (S13) As shown in Fig. 10, when the request is a reference request for an uncopied portion of the copy destination (access pattern 7 in Fig. 6), the processor 4 instructs that the target blocks be read from the copy-source logical disk 1, and performs staging of the target data blocks of the copy-source logical disk in the copy-source cache memory 7. Next, it copies the uncopied data blocks, that were read into the copy-source cache memory 7, to the copy-destination cache memory 9. The uncopied data blocks in the cache memory 9 are transferred to the host (application) 10. The processor 4 then instructs that the uncopied data blocks that

were copied into the cache memory 9 be written to the copy-destination logical disk 2, and then the uncopied data blocks that were copied into the cache memory 9 are written to the logical disk 2. The processor 4 sets the update block information for the copy-source bitmap 6 and copy-destination bitmap 8 to "0", indicating that update has been completed.

[0062] (S14) The processor 4 determines whether there are any data blocks of the referenced blocks still remaining. When there still are some data blocks of the referenced blocks remaining, the process returns to step S13. When there are no data blocks of the referenced blocks remaining, the reference process ends.

[0063] (S15) In normal reference processing, the processor 4 reads the specified reference blocks from the logical disk 1 or 2, and transfers them to the host. Then processing ends.

[0064] Moreover, the reference operation of copied portions of the copy source (access pattern 1 in Fig. 6) is as shown in Fig. 7. Also, the reference operation of uncopied portions of the copy source (access pattern 3 in Fig. 6) is as shown in Fig. 9. Furthermore, the reference operation of copied portions of the copy destination (access pattern 5 in Fig. 6) is the same as that shown in Fig. 7.

[0065] Next, the update process during copying will be explained with reference to Fig. 5.

[0066] (S20) When the processor 4 detects an update instruction (write), it then determines whether actual copying is in progress. When actual processing is not in progress, the process advances to the normal update processing of step S28.

[0067] (S21) When actual copying is in progress, then the processor 4 determines whether there is an update request for the copy source.

[0068] (S22) When there is an update request for the copy source, the processor 4 references the bitmap 6 and determines whether or not the requested update area is an uncopied portion. When the request is not an update request for uncopied portions of the copy source, or in other words, when the request is an update request for copied portions of the copy source (access pattern 2 in Fig. 6) the process advances to a normal update processing of step S28.

[0069] (S23) As shown in Fig. 11, when the request is an update request for uncopied portions of the copy source (access pattern 4 in Fig. 6), then the processor 4 instructs that the target update blocks, be read from the copy-source logical disk 1, and performs staging of the target update data blocks from the copy-source logical disk 1 in the copy-source cache memory 7. Next, it copies the target update data blocks, that were read into the copy-source cache memory 7, to the copy-destination memory 9. The processor 4 instructs that the target update data blocks that were copied into the cache memory 9 be written to the copy-destination logical disk 2. The target update data blocks that were copied into the cache memory 9 are then written to the logical disk

2. The processor 4 sets the update block information of the copy-source bitmap 6 and the copy-destination bitmap 8 to "0", indicating that update was completed. In addition, the processor 4 writes the update data blocks, that were transferred to the cache memory 7 from the host, to the logical disk 1.

[0070] (S24) The processor 4 determines whether there are any data blocks of the update data still remaining. When there are data blocks of update data still remaining, the process returns to step S23. When there are no data blocks of update data remaining, the update process ends.

[0071] (S25) When the request in step S21 is not an update request for the copy source, then it is an update request for the copy destination. The processor 4 references the bitmap 8 and determines whether or not the requested update area is an uncopied portion. When the request is not an update request for uncopied portions of the copy destination, or in other words, when the request is an update request for copied portions of the copy destination (access pattern 6 in Fig. 6), then the process advances to the normal update processing of step S28.

[0072] (S26) When the request is an update request for uncopied portions of the copy destination (access pattern 8 in Fig. 6), then, as shown in Fig. 12, the processor 4 instructs that the update data blocks, that are stored in the cache memory 9, be written. The update data blocks that are stored in the cache memory 9 are then written to the logical disk 2. The processor 4 sets the update block information of the copy-source bitmap 6 and the copy-destination bitmap 8 to "0", indicating that update has been completed. By doing this, copying of this area by the copy operation is prohibited.

[0073] (S27) The processor 4 determines whether there are data blocks of update data still remaining. When there are data blocks of update data remaining, the process returns to step S26. When there are no data blocks of update data, the update process ends.

[0074] (S28) In normal update processing, the processor writes the update data blocks, that were transferred to the cache memory 7 or 9 from the host, to the specified logical disk 1 or 2. The update process then ends.

[0075] The update operation of copied portions of the copy source (access pattern 2 in Fig. 6) is as shown in Fig. 8. Also, the update operation of copied portions of the copy destination (access pattern 6 in Fig. 6) is the same as shown in Fig. 8.

[0076] In summarizing the operations described above, there are a total of eight access patterns from the host during copying as shown in Fig. 6. In referencing of copied portions of the copy-source or copy-destination logical disks 1,2 (access patterns 1,5), data blocks of reference data from the specified logical disk 1 or 2 are staged in the cache memory 7, 9, and then the host is notified of the staged data blocks (see Fig. 7). This processing is influenced by the copied/uncopied

status of the data blocks, and is identical to normal reference processing.

[0077] In updating copied portions of the copy-source or copy-destination logical disk 1, 2 (access patterns 2, 6), data blocks are updated in the cache memory 7, 9 specified from the host and are written in the copy-source or copy-destination logical disk 1, 2 (see Fig. 8). In this process, the data blocks are copied, so the process is the same as normal update processing.

[0078] In referencing uncopied portions of the copy-source logical disk (access pattern 3), the data blocks of reference data are staged in the cache memory 7 from the copy-source logical disk 1, and then the host is notified of the staged data blocks (see Fig. 9). This process is influenced by the copied/uncopied status of the data blocks, and so is identical to normal reference processing.

[0079] In this process, copying can also be performed by writing the staged data blocks to the copy-destination logical disk 2. When doing this, it is also possible to perform only the referencing described above, and to perform copying later.

[0080] In updating uncopied portions of the copy-source logical disk 1 (access pattern 4), the target update data blocks of the copy-source logical disk 1 are written (copied) to the copy-destination logical disk 2. When doing this, as shown in Fig. 3D, the update bits of the bitmaps 6, 8, that control the copy state, are updated to the copied state, and copying of this data block area is set as being completed. Then, the update data blocks from the host are written to the copy-source logical disk 1 (see Fig. 11).

[0081] In referencing uncopied portions of the copy-destination logical disk 2 (access pattern 7), the target reference data blocks from the copy-source logical disk 1 are staged in the cache memory 9, then the host is notified of the staged data blocks. Also, the staged data blocks are written to the copy-destination logical disk 2 (see Fig. 10). As shown in Fig. 3D, when doing this, the update bits for the data blocks of the bitmaps 6, 8, that control the copy state, are updated to the copied state, and copying of this data block area is set as being completed.

[0082] In updating uncopied portions of the copy-destination logical disk 2 (access pattern 8), the update data blocks from the host are written to the copy-destination logical disk 2 (see Fig. 12). As shown in Fig. 3D, at that time, the update bits of the bitmaps 6, 8, that control the copying state, are updated to the copied state, and copying of this data block area is set as being completed. Therefore, copying from the copy source of that data block area to the copy destination is prohibited.

[0083] In this way, copying is performed by divided up the copy range into a plurality of areas, and copying in units of these area divisions. Therefore it is possible to receive an I/O request from the host and process that I/O request during the copying process. In addition, it is possible to immediately access a logical disk even when

a copy instruction is received.

[0084] Moreover, when performing update access of an uncopied area of the copy-source logical disk, the uncopied area of the copy-source logical disk is updated after it is copied to the copy-destination logical disk. Therefore, it is possible to prevent the copy data of the copy-destination logical disk from becoming the updated data due to the update access.

[0085] Furthermore, when performing reference access of an uncopied area of the copy-destination logical disk, the data of the corresponding area of the copy-source logical disk is read and transferred. Therefore, it is possible to provide copy data even when performing reference access of an uncopied area of the copy-destination logical disk. Also, it is possible to add the process of copying the data of the area read from the copy-source logical disk to the copy-destination logical disk, as in this embodiment. By doing this, it is possible to use the read data in the copying process.

[0086] Furthermore, when performing update access of an uncopied area of the copy-destination logical disk, the data of the corresponding area of the copy-destination logical disk is updated with update data, and since copying to that area is prohibited, it is possible to prevent the update data from being changed by copy data.

[0087] Also, since control information is used for controlling the copy state of each area by way of a bitmap or the like, it is easy to determine whether the area is a copied area or uncopied area. Moreover, it is possible to prevent recopying during access processing even when the access area is copied before copying another area. Similarly, it is possible to prevent replacing update data with copy data even when updating the copy-destination logical disk.

[0088] The control information is a bitmap so it is possible to control copying with a small amount thereof. There are separate bitmaps 6, 8 for each logical disk, however it is also possible to use one common bitmap for each logical disk.

[0089] Moreover, it is possible to use conventional access technology of using a cache memory when accessing or copying the logical disks 1, 2. In other words, in a system where part of the data of the physical disk is staged in a cache memory, the logical disk is accessed by first accessing the cache memory and finding the appropriate block data. When the appropriate block data are not staged in the cache memory, the block data are obtained by accessing the physical disk. The same is true when performing an update. The data in the cache memory are updated. This kind of conventional cache memory can be used. In this case, the logical disks 1, 2 comprise a physical disk and cache memory.

[0090] Besides the embodiment described above, the invention can be changed as follows.

(1) In the embodiment described above, the proc-

essor 4 of the disk controller 3 executes the copying process, however, it is also possible for the host of a server or the like to execute the copying process. Also, it is possible to do away with the disk controller 3 by having the host of a server or the like execute the copying and access processes.

(2) During reference access of an uncopied area of the copy-destination logical disk, the copy-source logical disk is referenced and copied to the copy-destination logical disk, however, it is also possible reference the copy-source logical disk and copy to the copy-destination logical disk separately.

[0091] The preferred embodiment of the present invention has been explained, however, the invention is not limited to this embodiment and can be embodied in various forms within the scope of the present invention.

[0092] In another form of this invention, the access step further comprises a step of referencing the target access area of the copy-source logical disk when performing reference access of the copy-source logical disk, and a step of referencing or updating the target access area of the copy-destination logical disk when accessing copied areas of the copy-destination logical disk.

[0093] In another form of this invention, the step of referencing an uncopied area of the copy-destination logical disk comprises a step of copying the target access area of the referenced copy-source logical disk to the copy-destination logical disk after the target access area of the copy-source logical disk has been referenced. By doing this, it becomes possible to copy and reference simultaneously.

[0094] In another form of the invention, the access step further comprises a step of updating the control information of the target access area after the target access area of the copy-source logical disk has been copied to the copy-destination logical disk.

[0095] In this way, it is possible for the control information to accurately indicate the copying condition through access processing even if copying is performed in advance.

[0096] In another form of the invention, the copying step comprises a step of referencing the control information that indicates the copying condition of each of the areas of the copy-source logical disk and setting the next copy area, a step of reading the target area of the copy-source logical disk in memory, a step of updating the control information after the target area has been read, and a step of writing the target area of the memory to the copy-destination logical disk.

[0097] By reading the target area in memory, it becomes possible to access the copy-source logical disk by updating the control information.

[0098] In another form of the invention, the copying step comprises a step of referencing a bitmap indicated by a flag that indicates the copying condition of each area and setting the next copy area, and a step of

updating the copy area flag of the bitmap.

[0099] Since the control information is expressed as a bitmap, it is possible to reduce the amount of memory occupied by the control information.

[0100] In another form of the invention, the copying step further comprises a step of notifying when copying is finished according to the copy instruction. By doing this, it is possible to immediately know when the access source can be accessed.

[0101] As explained above, the present invention has the following effects.

(1) The copy-source logical disk is divided up into a plurality of areas, and copying to the copy-destination logical disk is performed in units of area divisions. When there is an access request while copying is in progress, the copying operation is interrupted and the access operation is performed. Therefore, it is possible to receive the access request and perform reference access or update access of the copy-source and copy-destination logical disks while actual data is being copied.

(2) Also, when performing update access of an uncopied area of the copy-source logical disk, the target access area of the copy-source logical disk is updated after the target access area of the copy-source logical disk is copied to the copy-destination logical disk. Therefore, it is possible to copy the data before the update when there is a copy instruction.

(3) When performing reference access of an uncopied area of the copy-destination logical disk, the data of the corresponding area of the copy-source logical disk is read. Therefore, it is possible to reference copy data of that area even when the area is an uncopied area of the copy-destination logical disk.

(4) When performing update access of an uncopied area of the copy-destination logical disk, copying that area is prohibited after the copy-destination logical disk has been updated with the update data. Therefore, it is possible to prevent changes to the update data of that area of the copy-destination logical disk due to copying.

Claims

1. A copy method for copying data from a copy-source logical disk to a copy-destination logical disk according to a copy instruction comprising:

a step of copying, in units of each area divided in the copy-source logical disk, from said copy-source logical disk to said copy-destination logical disk;

a step of interrupting said copy process in response to an access request; and

a step of accessing said copy-source logical

disk or copy-destination logical disk according to said access request during copying.

2. The copy method of claim 1, wherein

said access step comprises:

a step of updating a target access area of said copy-source logical disk after said target access area has been copied to said copy-destination logical disk when said access request is a request for updating of uncopied areas of said copy-source logical disk;

a step of referencing said target access area of said copy-source logical disk when said access request is a request for referencing of uncopied areas of said copy-destination logical disk; and

a step of updating said target access area of said copy-destination logical disk and prohibiting copying of said target access area when said access request is a request for updating of uncopied areas of said copy-destination logical disk.

3. The copy method of claim 1 or 2 wherein

said copying step comprises:

a step of referencing control information that indicates a copying condition of said areas, and setting a next copy area; and

a step of updating the control information of said copy area after copying of said copy area has finished; and wherein

said access step comprises a step of referencing said control information and determining whether said target access area is a copied area or uncopied area.

4. The copy method of claim 2 or 3, wherein

said access step further includes:

a step of referencing the target access area of the copy-source logical disk when performing reference access of the copy-source logical disk; and

a step of referencing or updating the target access area of the copy-destination logical disk when accessing copied areas of the copy-destination logical disk.

5. The copy method of claims 2, 3 or 4, wherein

said step of referencing an uncopied area of the copy-destination logical disk comprises:

a step of copying the target access area of the referenced copy-source logical disk to the copy-destination logical disk after the target access area of the copy-source logical disk has been referenced.

6. The copy method of claims 3, 4 or 5, wherein

said access step further comprises a step of updating the control information of the target access area after the target access area of the copy-source logical disk has been copied to the copy-destination logical disk.

7. The copy method of claims 3, 4, 5 or 6, wherein

the copying step comprises:

a step of referencing the control information that indicates the copying condition of each of the areas of the copy-source logical disk and setting the next copy area,

a step of reading the target area of the copy-source logical disk to memory,

a step of updating the control information after the target area has been read, and

a step of writing the target area of the memory to the copy-destination logical disk.

8. The copy method of any of claims 3 to 7, wherein

the copying step comprises:

a step of referencing a bitmap indicated by a flag that indicates the copying condition of each area and setting the next copy area, and

a step of updating the copy area flag of the bitmap.

9. The copy method of any preceding claim, wherein

the copying step further comprises a step of notifying completion of a copy according to the copy instruction.

10. A disk storage system comprising:

at least one physical disk for one logical disk and another logical disk, and

a disk controller for accessing data of a specified logical disk according to an access instruction, and copying data, in units of each area divided in a copy-source logical disk, of said copy-source logical disk to a copy-destination logical disk according to a copy instruction; wherein

said disk controller interrupts said copy process when it receives said access instruction during said copy process and accesses said logical disk according to said access instruction.

11. The disk storage system of claim 10, wherein said disk controller updates a target access area of said copy-source logical disk after it has been copied to said copy-destination logical disk when said

instruction indicates to update access of an uncopied area of said copy-source logical disk,

references said target access area of said copy-source logical disk when said instruction indicates to reference access of an uncopied area of said copy-destination logical disk, and updates said target access area of said copy-destination logical disk, and prohibits copying of said target access area when said instruction indicates to update access of an uncopied area of said copy-destination logical disk.

12. The disk-storage system of claim 10 or 11, wherein

said disk controller comprises:

a memory for storing the control information that indicates the copying condition of said areas of said copy-source logical disk, and control circuit for referencing the control information stored in said memory and determines whether said target access area is a copied area or an uncopied area.

13. A program for copying data from a copy-source logical disk to a copy-destination logical disk according to a copy instruction, wherein

said program comprises:

information for copying data, in units of area divisions of said copy-source logical disk, to said copy-destination logical disk, information for interrupting said copying process in response to an access instruction, and information for performing an access process of said copy-source or copy-destination logical disk during said copying process.

14. A program for performing processes of claim 1.

15. A storage medium on which is stored the program according to claim 13 or 14.

FIG. 1

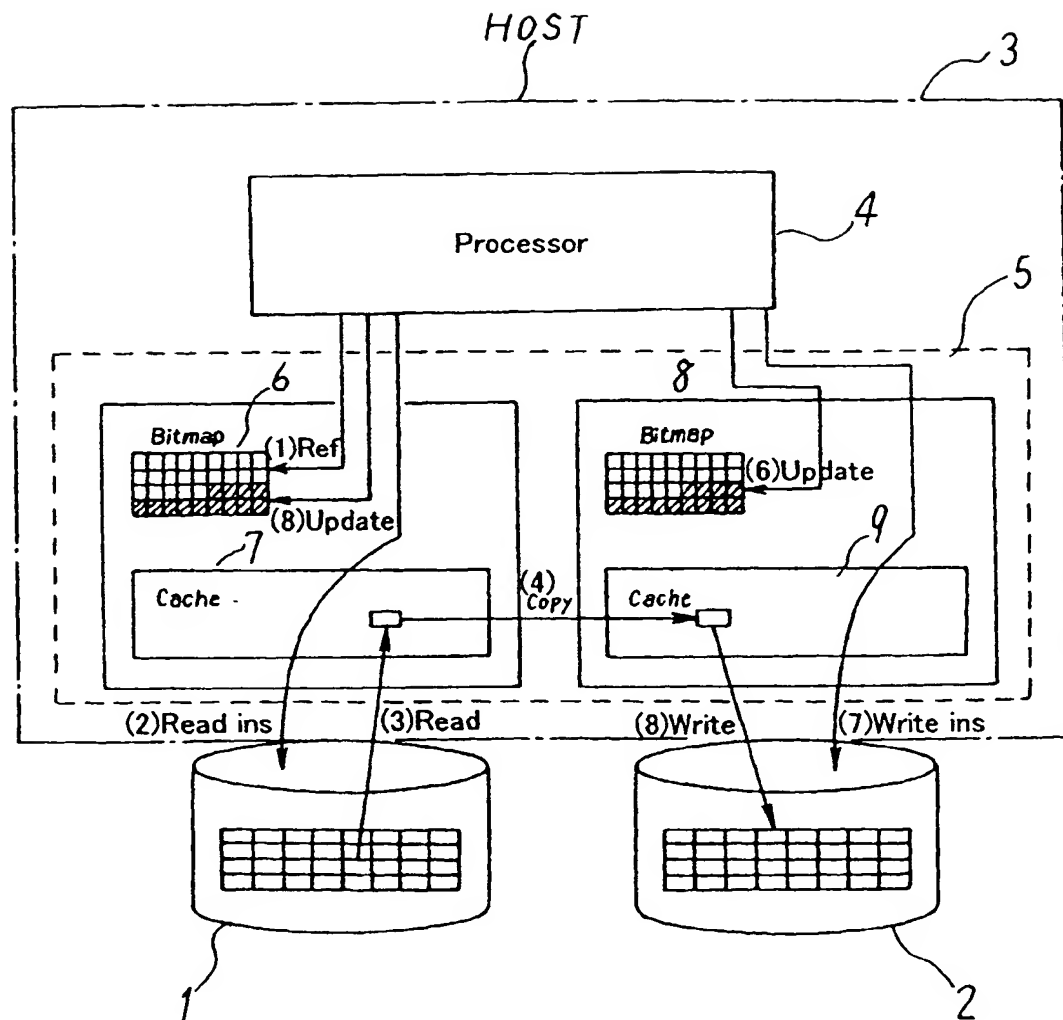


FIG. 2

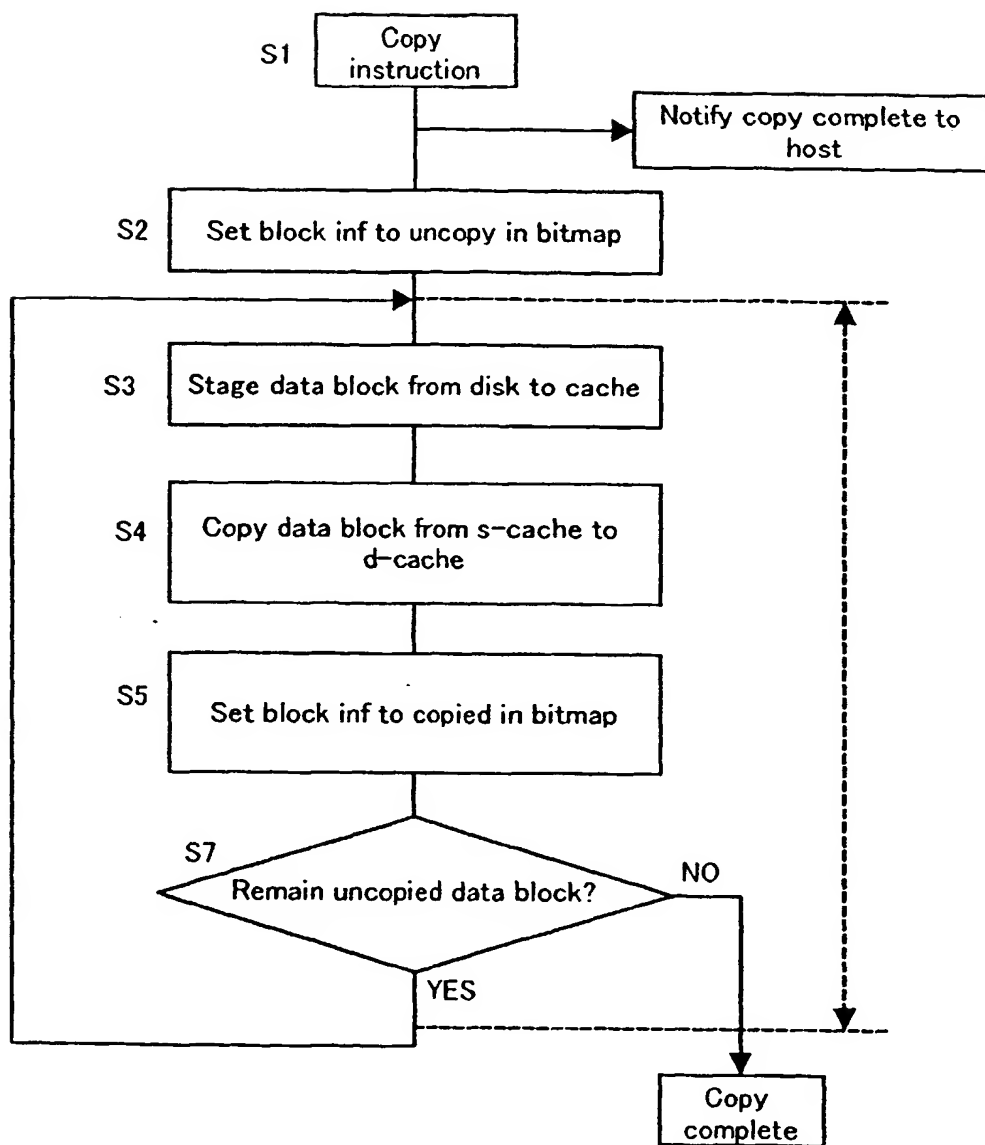


FIG. 3A

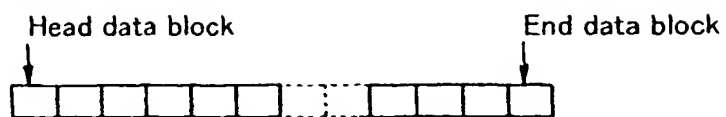


FIG. 3B



FIG. 3C

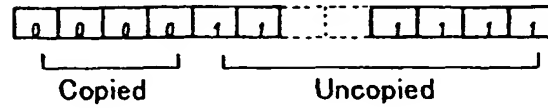


FIG. 3D

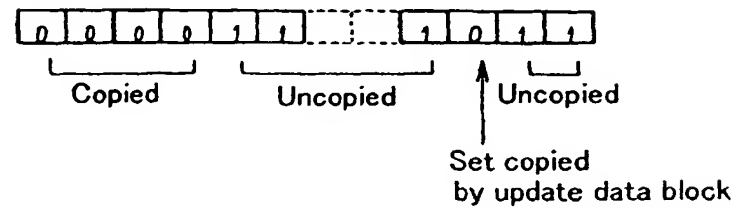


FIG. 4

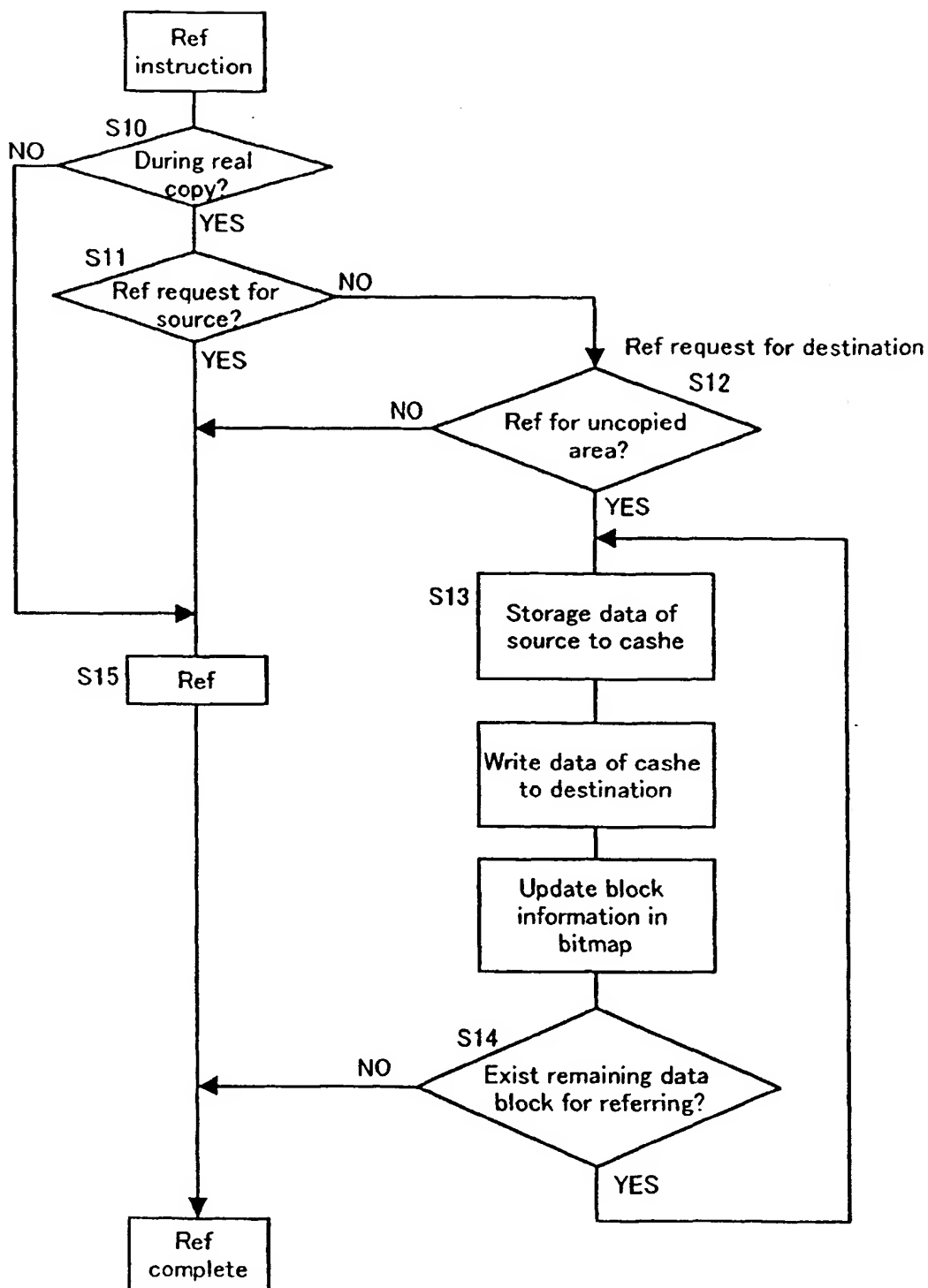


FIG. 5

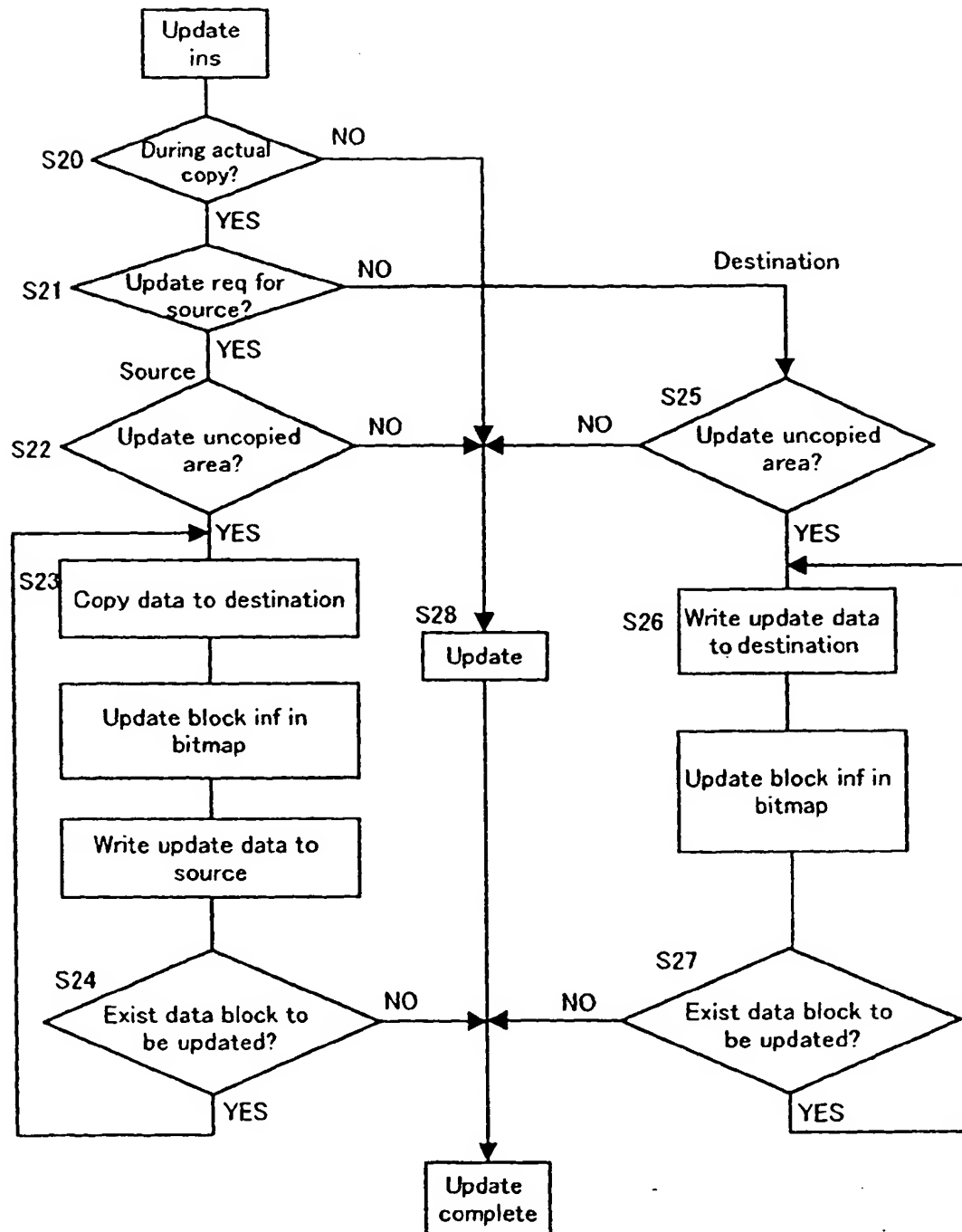


FIG. 6

Access pattern	Access logical disk	Actual copy	I/O access
Pattern 1	Source	Complete	Ref
Pattern 2			Update
Pattern 3		Uncomplete	Ref
Pattern 4			Update
Pattern 5	Destination	Complete	Ref
Pattern 6			Update
Pattern 7		Uncomplete	Ref
Pattern 8			Update

FIG. 7

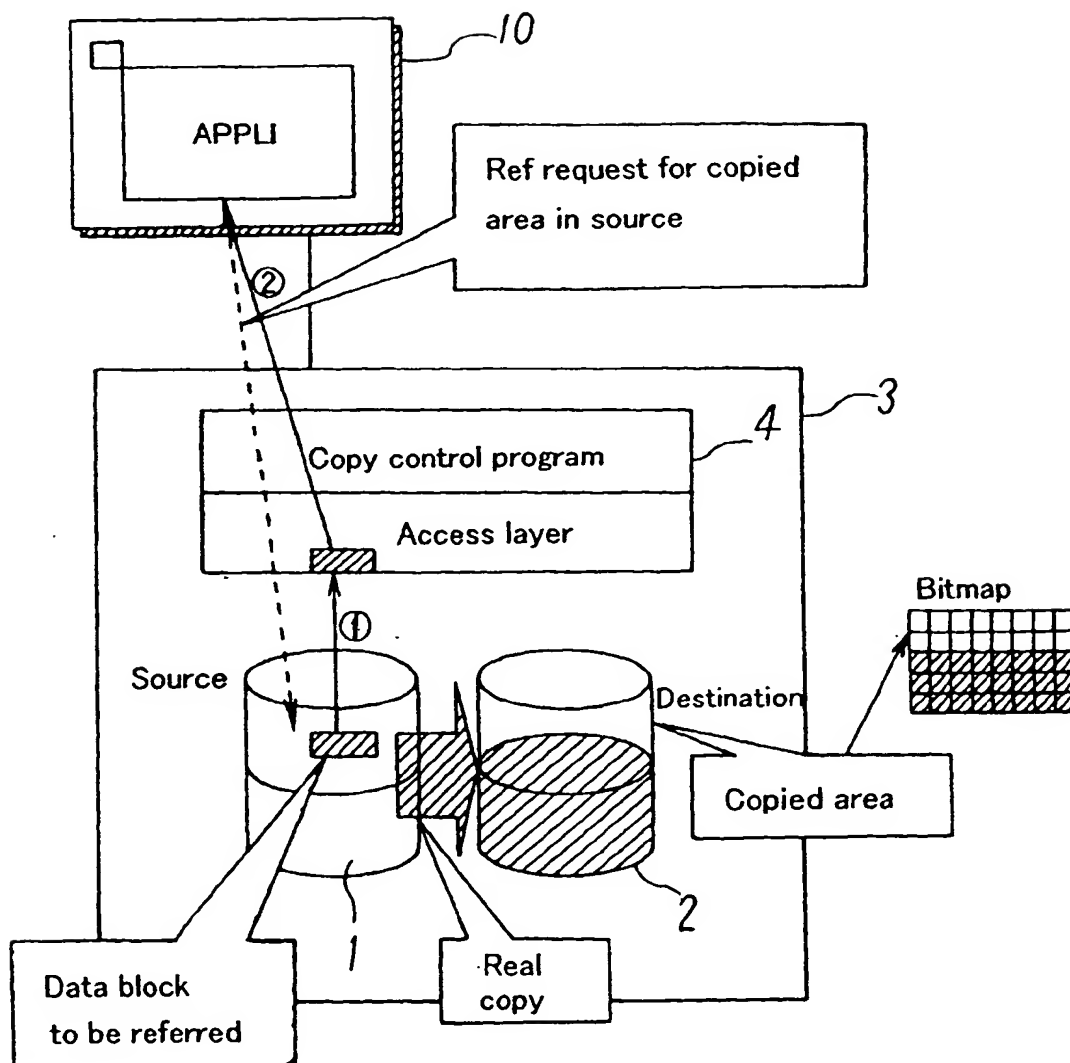


FIG. 8

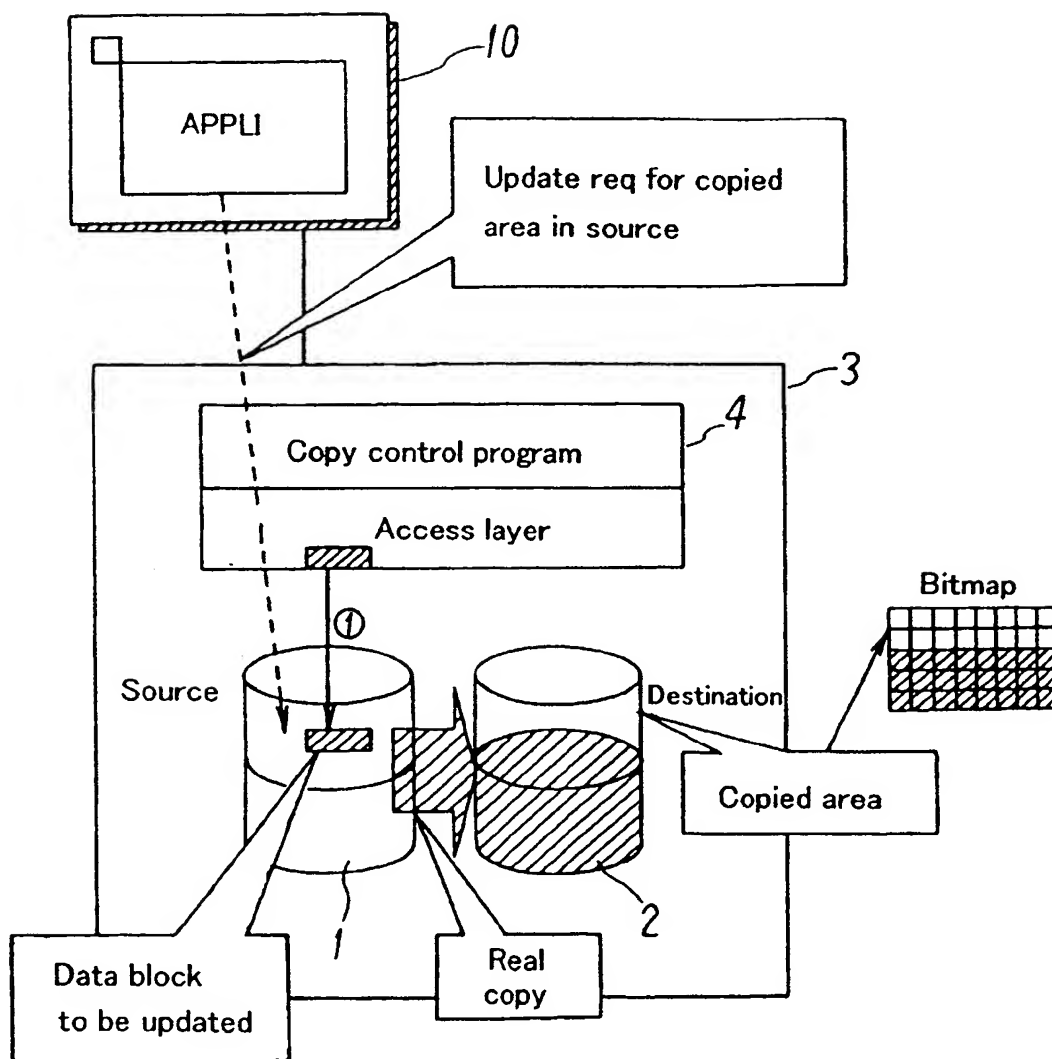


FIG. 9

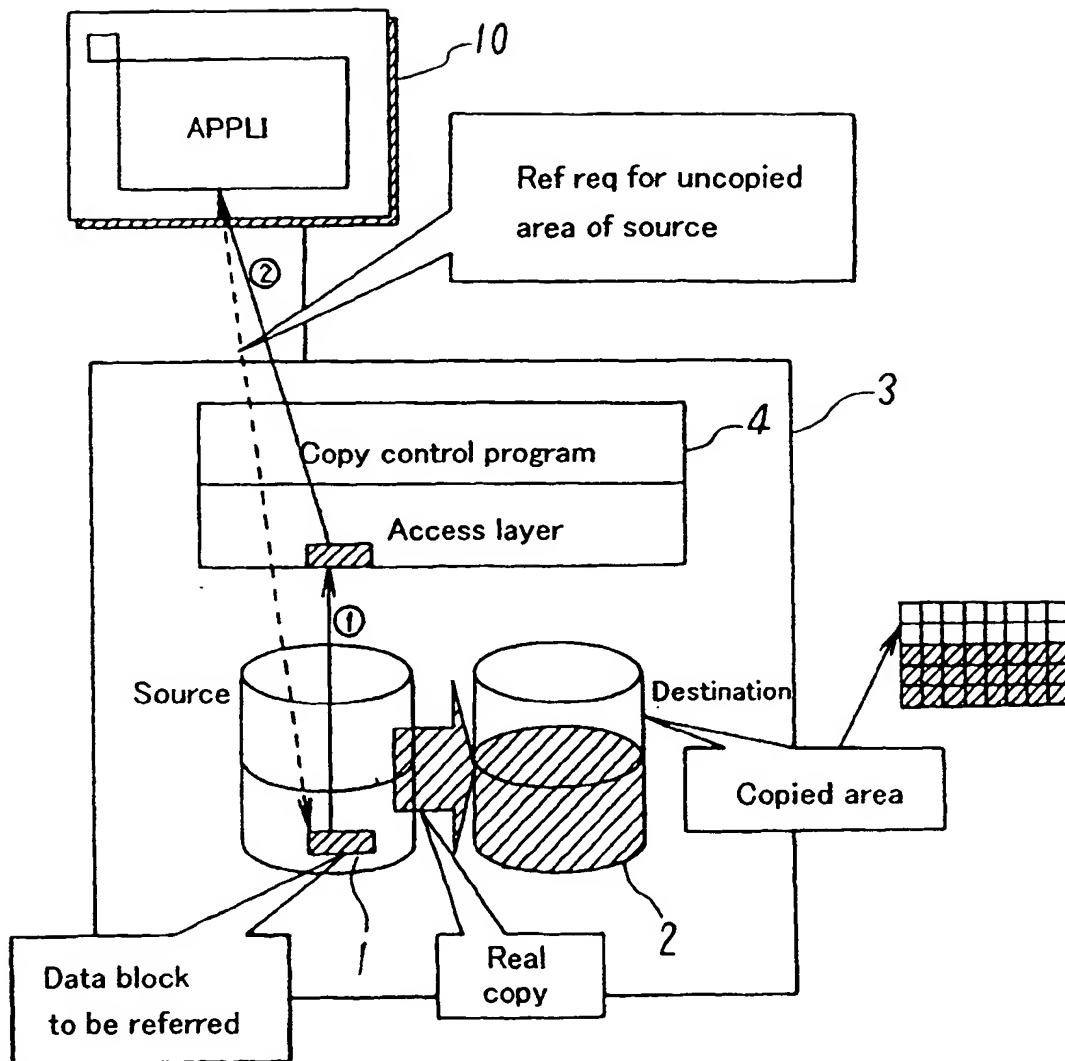


FIG. 10

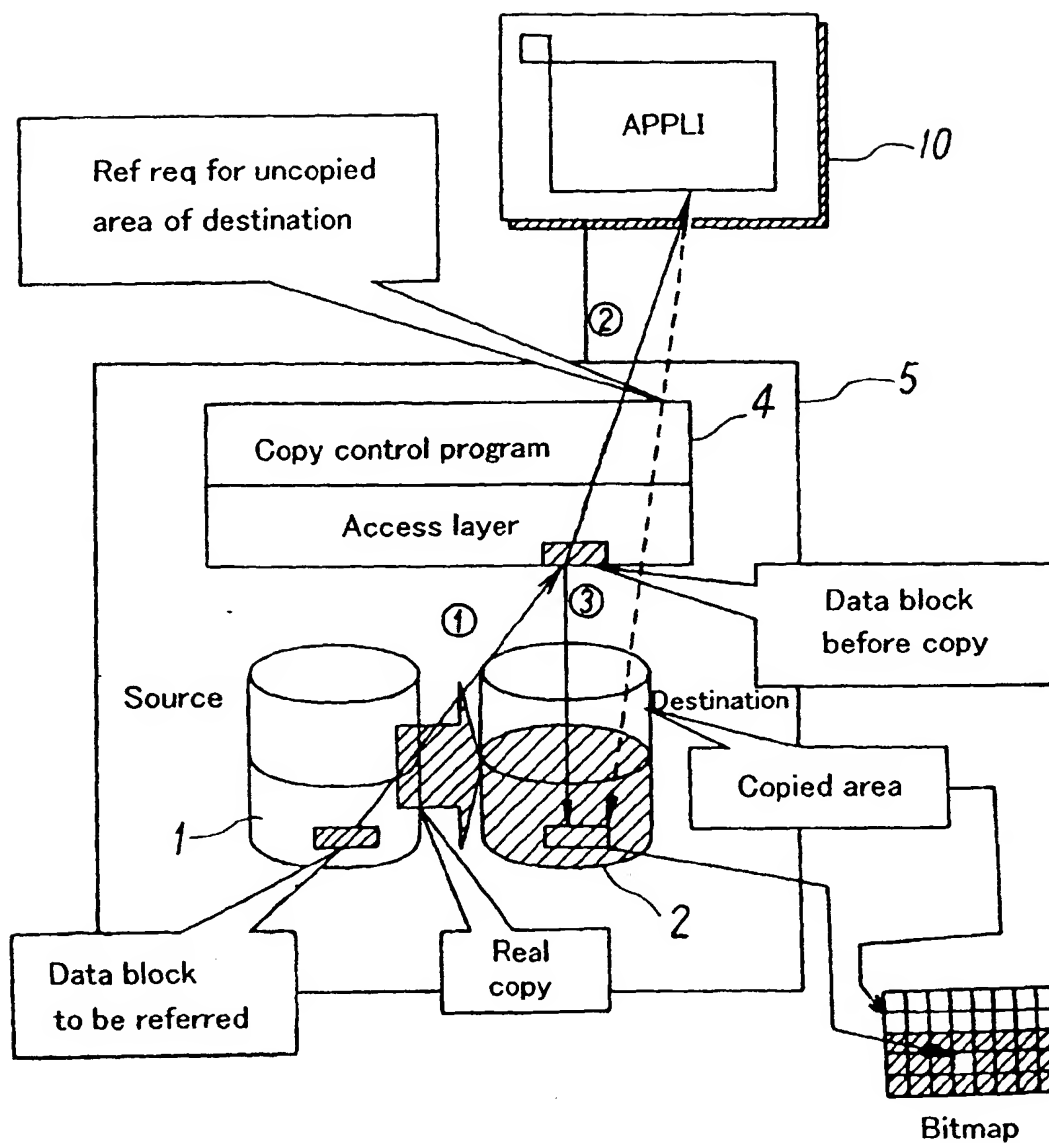


FIG. 11

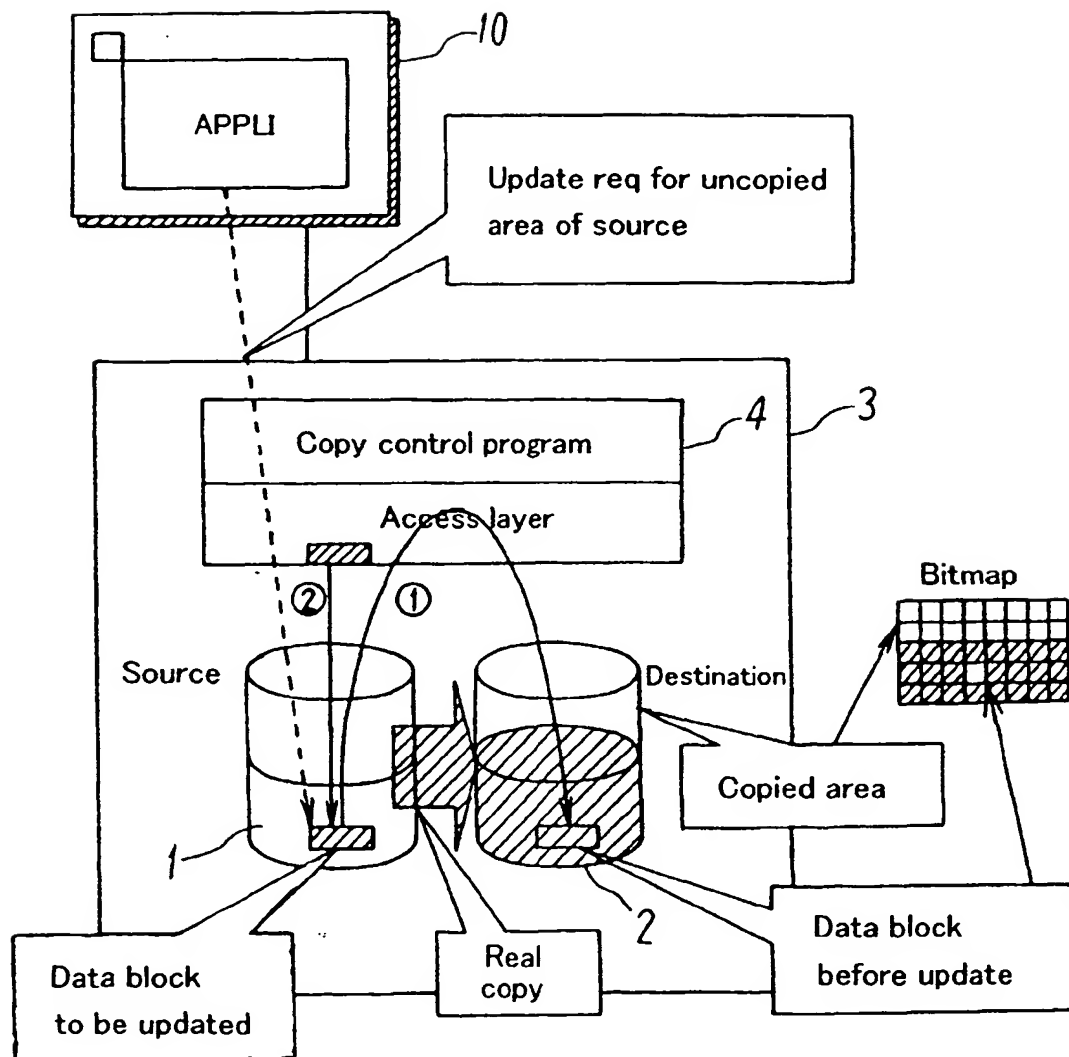


FIG. 12

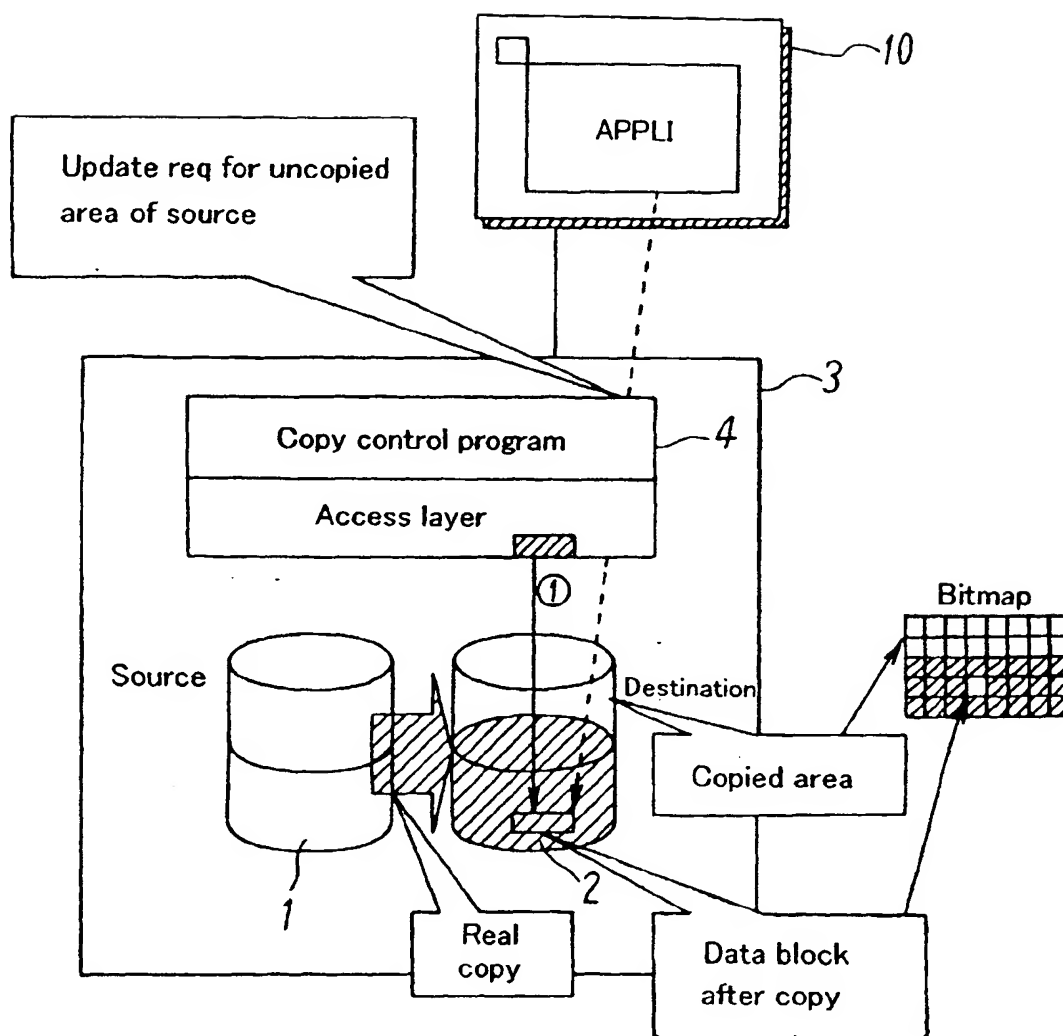


FIG. 13

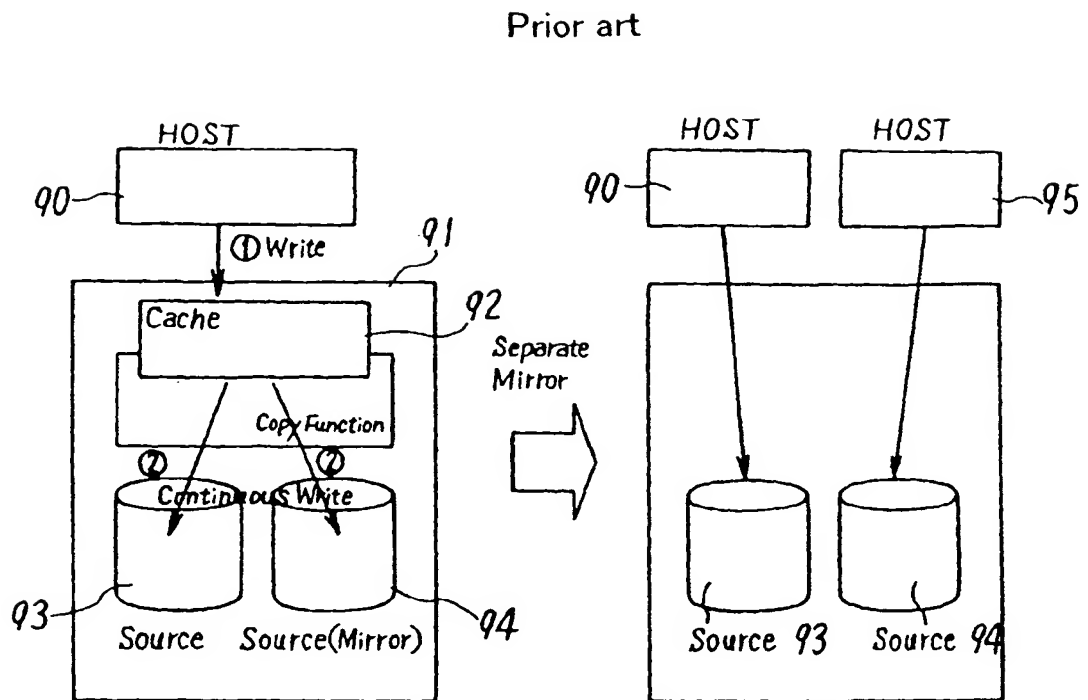


FIG. 14

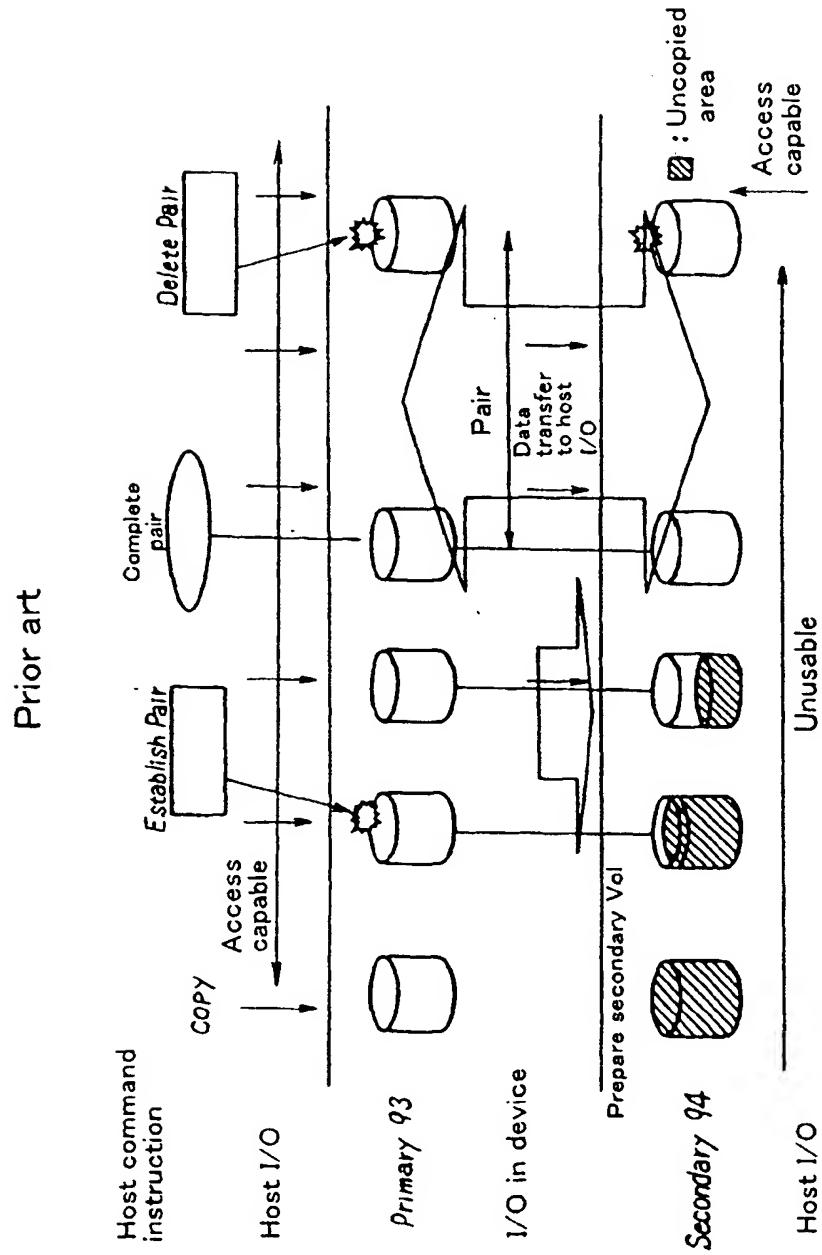


FIG. 15

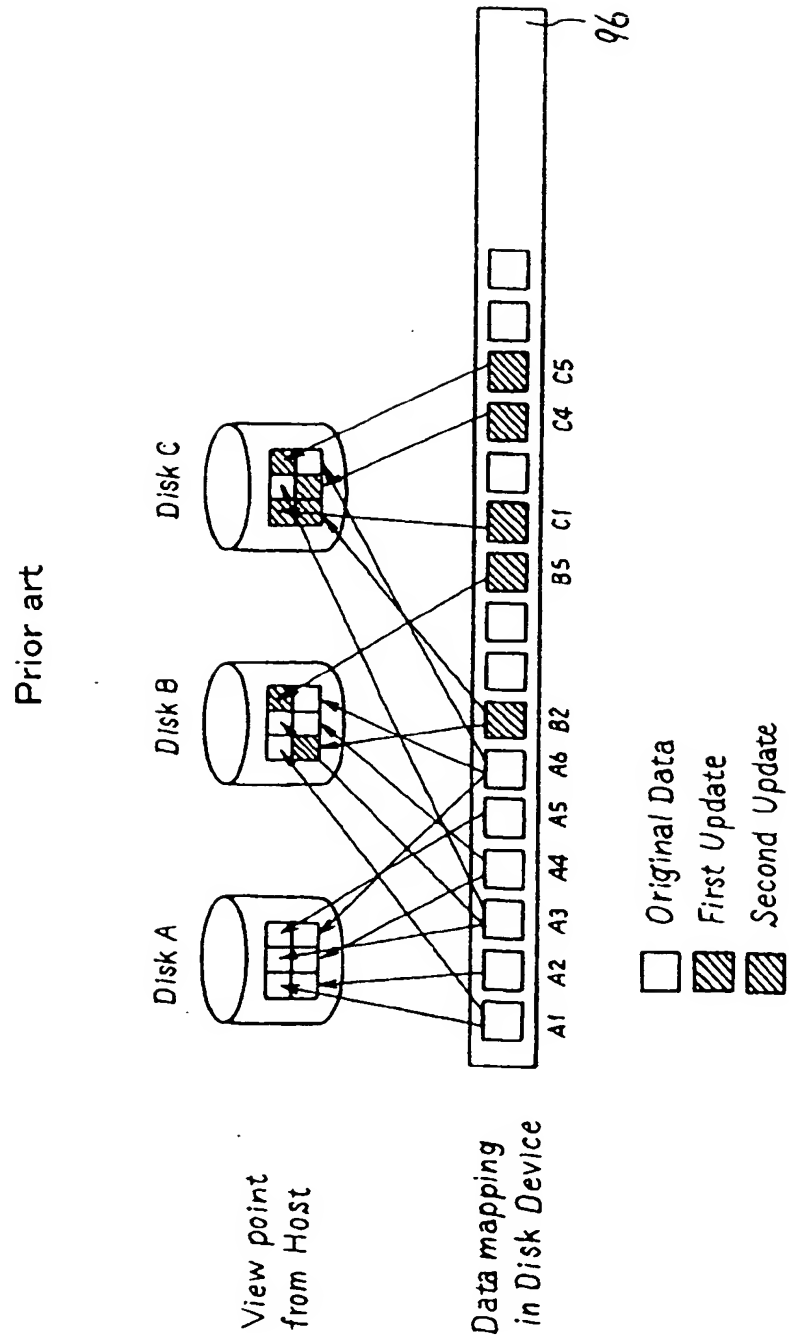
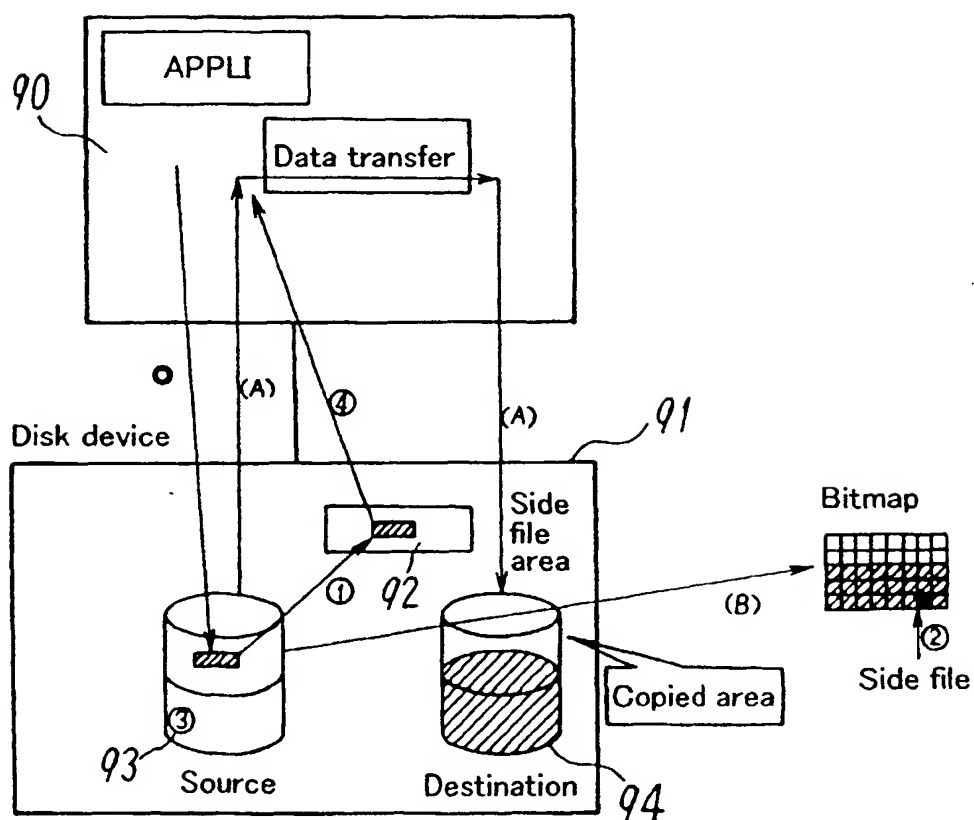


FIG. 16

Prior art





(11) **EP 1 091 283 A3**

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
08.05.2002 Bulletin 2002/19

(51) Int Cl.7: **G06F 3/06, G06F 11/14**

(43) Date of publication A2:
11.04.2001 Bulletin 2001/15

(21) Application number: 00305256.0

(22) Date of filing: 21.06.2000

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

- Onda, Yasuhiro, c/o Fujitsu Aichi Eng. Limited, Aichi 461-0004 (JP)
- Kato, Tadaomi, c/o Fujitsu Limited, Kawasaki-shi, Kanagawa, 211-8588 (JP)

(30) Priority: 30.09.1999 JP 27967899

(74) Representative: **Stebbing, Timothy Charles et al**
Haseltine Lake & Co.,
Imperial House,
15-19 Kingsway
London WC2B 6UD (GB)

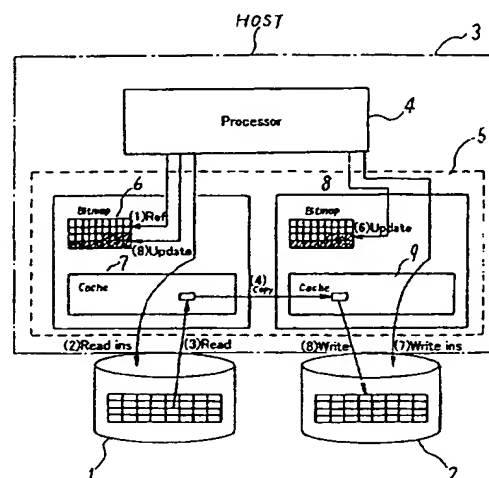
(71) Applicant: FUJITSU LIMITED
Kawasaki-shi, Kanagawa 211-8588 (JP)

(72) Inventors:
• Kaiya, Norikazu, c/o Fujitsu Aichi Eng.Limited,
Aichi 461-0004 (JP)

(54) Copying method between logical disks, disk-storage system and program for the same

(57) This invention relates to a copying method, disk storage system and program for copying data from one logical disk to another logical disk, and makes immediate access in response to a copy command possible. Copying from logical disk (1) (copy source) to logical disk (2) (copy destination) is performed for each area in the copy range, and when there is access, copying is interrupted. When performing update access of an uncopied area in the copy source (1), the area is copied from the copy source (1) to the copy destination, and then the area is updated (S23). When performing reference access of an uncopied area in the copy destination (2), the corresponding area in the copy source (1) is referenced. When performing update access of an uncopied area in the copy destination (2), that area in the copy destination is updated, and copying is prohibited (S13). In this way, it is possible to access the copy source (1) and copy destination (2) while copying is in progress, and it is possible to immediately regard copying as completed by a copy instruction.

FIG. 1





European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 00 30 5256

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	WO 97 24668 A (IPL SYSTEMS INC.) 10 July 1997 (1997-07-10) * page 3, line 19 - page 4, line 14 * * page 5, line 8 - page 9, line 25; figures 2A,2B *	1-4,6-12	G06F3/06 G06F11/14
A	--- PATENT ABSTRACTS OF JAPAN vol. 017, no. 557 (P-1626), 7 October 1993 (1993-10-07) & JP 05 158625 A (HITACHI LTD.), 25 June 1993 (1993-06-25) * abstract *	1,10	
A	--- US 5 799 147 A (SHANNON) 25 August 1998 (1998-08-25) * column 2, line 43 - column 3, line 37 * -----	1,10	
			TECHNICAL FIELDS SEARCHED (Int.Cl.7)
			G06F
The present search report has been drawn up for all claims			
Place of search BERLIN		Date of completion of the search 13 March 2002	Examiner Taylor, P
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

EPO FORM 1503 03.82 (P04C01)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 00 30 5256

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

13-03-2002

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 9724668	A	10-07-1997	WO 9724668 A1	10-07-1997
JP 05158625	A	25-06-1993	NONE	
US 5799147	A	25-08-1998	US 5852713 A	22-12-1998

EPO FORM P0456

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

This Page Blank (uspto)